Dzięki firmom polonijnym i rzemieślniczym zwiększa się na polskim rynku liczba użytkowanych mikrokomputerów. Chociaż firmy oferujące sprzęt do zastosowań profesjonalnych gwarantują serwis w terminie do 48 godzin od chwili zgłoszenia uszkodzenia, to jednak gdy utrzyma się tempo instalowania nowego sprzętu i równocześnie przybędzie niefachowych użytkowników, mogą powstać spore problemy. Odbiorców sprzętu z państwowych wytwórni też nie możemy skwitować krótkim: sami są sobie winni. Wreszcie niektóre „groźne" awarie często wynikają z braku doświadczenia użytkownika. Być może zamieszczony tekst zaoszczędzi tego rodzaju kłopotów przynajmniej niektórym naszym Czytelnikom.

Mikrokomputerowa pierwsza pomoc

Człowiek jest omylny, jednak skutki jego błędów mogą okazać się niewielkie w porównaniu do tych, jakie może spowodować „pomylony" komputer. O ile człowiek myli się zazwyczaj wykonując obliczenia, o tyle uszkodzenie komputera może prowadzić do utraty ogromnych ilości cennych informacji i to niemal w mgnieniu oka. Z tego właśnie powodu powszechnie zaleca się możliwie częste sporządzanie kopii zbiorów roboczych. Ta pozornie mało twórcza czynność powinna być powtarzana przynajmniej po każdym dniu pracy na komputerze.

Mikrokomputery to w sumie dosyć skomplikowane maszyny i zazwyczaj psują się w wyrafinowany sposób. Dlatego też trudno sporządzić wyczerpującą listę możliwych usterek. W poniższym tekście spróbujemy wspomnieć o kilku najbardziej popularnych. Nie będziemy oczywiście pisać w tym miejscu jak zastąpić serwis komputerowy, jednak w niektórych sytuacjach faktycznie możemy sobie sami poradzić lub przynajmniej skonkretyzować nasze „pretensje" w stosunku do komputera. To ostatnie jest o tyle ważne, że z chwilą przybycia serwisu komputery mają zwyczaj złośliwie udawać, że wszystko jest w porządku. Sporządzona na drukarce kopia naszych działań może okazać się bardzo pomocna w zlokalizowaniu uszkodzenia. Nie chodzi tu tylko o zaoszczędzenie czasu osoby naprawiającej, ale też w dużej mierze o zaoszczędzenie naszych nerwów przy wielokrotnym wzywaniu serwisu do tego samego uszkodzenia.

Niestety dosyć często zdarza się, że fabrycznie nowe komputery nie funkcjonują poprawnie. Dlatego też warto upierać się, aby instalacja i sprawdzenie systemu (zazwyczaj wpisywane do rachunku) faktycznie zostało wykonane. Nowemu komputerowi warto na początku poświęcić trochę czasu, choćby wprowadzając przykładowe procedury zamieszczane w instrukcji obsługi. Nie należy przy tym przeceniać własnego braku doświadczenia. Jeśli uważnie przeczytaliśmy komentarz do programu i mamy pewność, że wprowadzony program jest identyczny z tym, co wydrukowano w instrukcji, a mimo to nie osiągamy spodziewanego rezultatu, to istnieje duże prawdopodobieństwo, że system nie funkcjonuje prawidłowo.

Zakłócenia elektrostatyczne nie są na szczęście zbyt często występującą bolączką. Jeżeli jednak po kilkukrotnym przejściu pokoju, w którym pracuje komputer, powodujemy wyładowanie elektrostatyczne w zetknięciu z kaloryferem, to możemy spodziewać się kłopotów. Najczęściej objawiają się one nieoczekiwaną zmianą wyświetlanej na monitorze informacji lub też utratą kontroli nad systemem. Zazwyczaj wyzerowanie systemu (ang. RESET) przywraca mu pełnię sił witalnych. Środki zaradcze mogą być najróżniejsze: poczynając od zmiany umeblowania (likwidacja plastikowej wykładziny, inny stół etc.), a kończąc na zmianie stroju operatora, który powinien unikać ubiorów z włókien sztucznych.

Zakłócenia napięcia zasilającego

Różne typy komputerów mają różną odporność na niespodzianki z gniazdka sieciowego. Komputerom rodem z Wielkiej Brytanii często nie podoba się, że w sieci zamiast przepisowych 220 V jest niewiele ponad 200. Krótkie nawet przerwy w napięciu zasilania zazwyczaj kończą się utratą kontaktu z komputerem lub przynoszą efekt identyczny jak zerowanie. Osobną klasą zakłóceń są przepięcia generowane przez innych odbiorców energii elektrycznej. Należy oczekiwać, że znakomita większość Czytelników potrafi przewidzieć konsekwencje zasilania z jednego gniazdka komputera i grzejnika z termostatem. Jeżeli jednak komputer musi pracować w „uprzemysłowionej" okolicy, to należy pomyśleć o kupnie dobrego stabilizatora. Jak wskazuje jego nazwa powinien on także poradzić sobie ze zbyt niskim napięciem w sieci. Jeżeli mimo stabilizatora zakłócenia nadal wyprowadzają komputer z równowagi, to mamy dwie możliwości: poszukać innego typu komputera, z lepiej rozwiązanym zasilaczem lub też zamienić pomieszczenie w puszkę Faraday'a. To drugie rozwiązanie, choć kosztowne, często bywa mało skuteczne.

Problem nie włączonej drukarki

Proszę się nie śmiać, tego typu błąd zdarza się wcale nie tak rzadko nawet wieloletnim użytkownikom komputerów. Bardzo często zdarza się, że serwis przyjeżdża, aby stwierdzić... że skończył się założony na drukarkę papier. Zazwyczaj gdy komputer dochodzi do fragmentu programu realizującego wydruk, który z jakiś powodów nie może zostać wykonany, to potulnie czeka aż operator włączy ją do sieci, przełączy w tryb „on line" lub wetknie kabel połączeniowy.

Z problemem tym można sobie stosunkowo łatwo poradzić nawet wtedy, gdy interfejs z drukarką jest maksymalnie uproszczony. Należy przed każdą procedurą przesyłającą dane na drukarkę umieścić krótki program. Powinien on wyświetlać na ekranie monitora tekst: „Włącz drukarkę, ośle!", a następnie wysyłać na drukarkę kod przejścia do nowej linii (LF — w języku BASIC instrukcją LPRINT bez parametrów). W następnej linii programu należy zamieścić polecenie wymazania napisu na monitorze i dalej sterowanie można już przekazać do procedury drukowania. W ten prosty sposób, ilekroć „program „zawiesi się" na niegotowej drukarce, dostaniemy od komputera odpowiednią reprymendę. Gdy drukarka będzie gotowa napis będzie zbyt krótko na ekranie, aby można było cokolwiek odczytać.

Uszkodzenia monitora

Typowe uszkodzenia to „pływanie" obrazu lub brak odchylania (pozioma jasna kreska). Całkowity brak obrazu jest uszkodzeniem stosunkowo rzadkim i z reguły poprzedzonym gwałtownym rozbłyśnięciem, trzaskiem lub obłoczkiem dymu. Brak obrazu najczęściej spowodowany jest złym połączeniem z komputerem lub też awarią samego komputera. Uszkodzony lub niepodłączony monitor zazwyczaj nie blokuje realizacji programu. W mikrokomputerach wyższej klasy, wyposażonych w drukarkę możemy przeprowadzić stosunkowo prosty test: należy ściągnąć system z dyskietki (MS-DOS, PC-DOS lub CP/M), wprowadzić z klawiatury CTRL-P i podać przykładowo komendę DIR. Wydruk znajdujących się na dyskietce programów będzie sugerował, że komputer jest w porządku.

Uszkodzenia pamięci RAM

Uszkodzenia tego typu objawiają się zazwyczaj od początku użytkowania mikrokomputera lub zaraz po dołączeniu bloku rozszerzającego obszar pamięci. Niektóre układy pamięci zaczynają wadliwie funkcjonować w podwyższonej temperaturze, a więc dopiero po pewnym okre-

sie pracy komputera. Błędy pamięci bywają często trudne do wykrycia. Częstym symptomem jest fakt, że niektóre programy pracują prawidłowo, podczas gdy inne „padają" od początku lub powodują zawieszenie systemu w trakcie realizacji. Jeżeli uszkodzenie pojawi się w obszarze przeznaczonym na pamięć obrazu, to zazwyczaj widać je w postaci ciemnych punktów lub dziwnych zjaw na ekranie. Jeżeli uszkodzenie trafiło się w obszarze, w którym przechowywane są ważne zmienne systemowe, to po wyzerowaniu komputera na całym ekranie pojawią się dziwaczne stwory, a w tle czasem paski lub nawet krata. W tym przypadku uszkodzenie nie budzi wątpliwości — objawy złego funkcjonowania są „stabilne".

Najwięcej kłopotu sprawiają uszkodzenia pojedynczych bitów pamięci RAM w obszarze, w który ładowany jest system operacyjny lub program użytkowy. Przez stosunkowo długi czas uszkodzenie może zupełnie nie objawiać się, gdyż akurat nie było odwołań do fragmentu kodu znajdującego się w tym miejscu. Jeśli jednak już natrafimy na „chore" miejsce, to prawie na pewno zakończy się to utratą kontaktu z komputerem. Wyświetlany na ekranie obraz może się zmienić lub nie, ale komputer nie będzie reagował na przyciskane klawisze.

Do wykrywania błędów tego typu służą specjalne programy nazywane testami pamięci. Najprostszy test można napisać posługując się nawet językiem BASIC. Należy najpierw instrukcją POKE zapisać zerami cały obszar pamięci, przeznaczony na programy użytkownika (z wyjątkiem miejsca, gdzie znajduje się sam program testu pamięci), a następnie odczytać wszystkie lokacje, sprawdzając, czy mają wartość zero. Potem należy zapisać pamięć wartością 255 (OFF_H) i wykonać sprawdzenie. Całość warto powtórzyć kilka razy, mimo że czas wykonywania tego typu programu jest stosunkowo długi. Jeżeli nie dysponujemy interpretem języka BASIC lub innym programem narzędziowym, który pozwoliłby nam zrealizować opisany algorytm (w komputerach zorientowanych na określone zastosowania, np. biurowych, jest to całkiem prawdopodobne!), możemy posłużyć się inną metodą. Zadowalające wyniki może dać program elektronicznego formularza rachunkowego typu VisiCalc. Należy po prostu zdefiniować poszczególne pola w ten sposób, aby w każdej komórce była wykonywana operacja polegająca na zwiększeniu o jeden wartości z poprzedniej komórki. Kolejne pola należy definiować aż do wyczerpania przeznaczonego na ten cel obszaru pamięci RAM. Gdy proces definiowania zostanie zakończony, należy ponawiać zlecenie przeliczenia wartości dopóki komputer nie „zatka się". Przykładowo dla APPLE i programu VisiCalc przeliczenie może być wykonane 32 razy. Jeżeli w obszarze wykorzystywanym przez program istnieje uszkodzenie, to końcowy wynik kalkulacji nie będzie prawidłowy. Wykonanie pojedynczego przeliczenia nie gwarantuje jeszcze, że uszkodzony bit nie będzie przypadkowo ustawiony na wartość, jaką powinien mieć faktycznie. Wielokrotne powtórzenie przeliczania zmniejsza prawdopodobieństwo takiego przypadku niemal do zera.

Bezsprzeczną wadą badania pamięci poprzez niekonwencjonalne wykorzystanie programu elektronicznego formularza rachunkowego jest kłopot z określeniem lokacji, która uległa uszkodzeniu. Opisany wcześniej algorytm prostego programu testującego pamięć można bez trudu przystosować tak, aby wyświetlał adres komórki, w której stwierdzona została nieprawidłowość.

Pozostał nam jeszcze jeden element systemu, wyjątkowo podatny na uszkodzenia. Chodzi tu o pamięć masową na dyskach elastycznych. Ale o tym — innym razem.

AJP

Na komfort pracy z mikrokomputerem składa się między innymi szybkość, z jaką realizuje on nasze zlecenia. Cytowane w tekście testy ograniczają się niestety tylko do pewnego wycinka rzeczywistości, jakim jest interpreter języka BASIC. Podobne testy można by wymyślić i dla innych języków — jednak mikrokomputery wyposażone są „firmowo" zazwyczaj tylko w BASIC. Interpretery lub kompilatory innych języków to najczęściej oferta niezależnych producentów oprogramowania, co najwyżej przystosowana dla danego typu mikrokomputera. Uniwersalnych metod testowania innych cech komputerów (np. szybkości współpracy z pamięcią masową) na razie nie opracowano i raczej nieprędko to nastąpi. Wynika to ze znacznej liczby różnych sposobów realizacji typowych wydawałoby się — funkcji.

Siedem z cytowanych w artykule testów publikowaliśmy już w zeszłym roku. Wielu Czytelników zwróciło nam jednak uwagę, że znacznie częściej wykorzystywany jest zestaw złożony z ośmiu testów. Ponieważ ósmy test dotyczy stosunkowo ważkiej cechy — szybkości wyliczania funkcji, zdecydowaliśmy się na jego publikację. Mając na uwadze nowych Czytelników i znane trudności ze zdobyciem INFORMATYKI, drukujemy cały zestaw.

Uwaga: autor poniższego artykułu przetłumaczył z języka angielskiego na język polski m.in. komentarz umieszczany w linii 100. W oryginale użyte jest słowo „BENCHMARK" — nie ma to jednak praktycznego wpływu na mierzony czas. Początkującym Czytelnikom zwracamy uwagę, że niektóre interpretery języka BASIC wymagają pełnego zapisu po słowie kluczowym THEN. Numer linii należy wtedy poprzedzić instrukcją GOTO.

Wyścig z czasem

Brytyjski miesięcznik PERSONAL COMPUTER WORLD (PCW) regularnie publikuje omówienia nowo pojawiających się na rynku mikrokomputerów. Jedną z badanych cech jest szybkość wykonywania podstawowych instrukcji języka BASIC. PCW stosuje w tym celu zestaw ośmiu krótkich programów testowych — ten sam od założenia pisma, tj. od 1978 roku. Umożliwia to porównywanie wyników testów dla różnych komputerów.

Styczniowy numer tego czasopisma przynosi bieżącą listę 68 mikrokomputerów o najszybszych interpreterach języka BASIC. Pierwsze miejsce na tej liście zajmuje — już od dwóch lat — rewelacyjny **SAGE II**, znany też z oryginalnej metody, jaką zastosowano przy jego projektowaniu. Na drugim miejscu jest mniej znany i znacznie nowszy **CRYSTAL**. Oba te urządzenia zawierają mikroprocesor MOTOROLA 68000 oraz, co chyba w tym wypadku ważniejsze, realizują częściowo skomplikowaną wersję języka BASIC, tzw. kod pośredni. Nieznacznie wolniejszy jest trzeci na liście **IBM AT**, zbudowany na nowym mikroprocesorze INTEL 80286.

Uznawany za największego konkurenta rodziny IBM PC — **APPLE MACINTOSH** zajmuje dopiero 12 miejsce, jednakże głównie ze względu na wyjątkowo wolną, jak na tę klasę maszyn, realizację funkcji. Jeśli pominiemy wyniki testu nr 8, pozwalającego ocenić prędkość wyliczania wartości funkcji sinus i logarytm, okaże się, że średnia z pozostałych siedmiu testów jest niemal identyczna, jak dla maszyn IBM AT i OLIVETTI M 24. Oznacza to mniejszą przydatność MACINTOSHA dla wszystkich tych zastosowań, gdzie wykonywane są skomplikowane obliczenia matematyczne, do czego nie był on jednak projektowany [1]). Za to najstarszym na liście i wcale nie ostatnim jest słynny **APPLE II.**

Najszybszym mikrokomputerem 8-bitowym okazał się mało znany **JONOS**, używający również bardzo szybkiego

---

[1]) Sąd autora tekstu jest może nazbyt pochopny. W zagadnieniach numerycznych, choć pożądana jest szybkość, to jednak istotniejsza jest często dokładność numeryczna. Warto dodać, że MACINTOSH zdobył sobie dużo zwolenników na amerykańskich uniwersytetach — raczej nie ze względu na piktogramy i myszkę (przyp. AJP).

mikroprocesora Z80B. Wysoką pozycję zajmuje też trzyletni już BBC B firmy ACORN. Wydaje się, że brytyjskie ministerstwo oświaty dokonało właściwego wyboru zalecając w swoim czasie ten solidny i szybki model dla szkół średnich i podpisując odpowiednią umowę z producentem.

Bardzo szybki jest też coraz popularniejszy AMSTRAD CPC 464. Wydaje się, że może on w najbliższym czasie stać się najpopularniejszym brytyjskim mikrokomputerem domowym. Jest on nie tylko szybki, ale również solidnie wykonany i stosunkowo tani. Dostępnością oprogramowania przewyższa on wszystkich potencjalnych konkurentów. Stosunkowo nisko plasuje się natomiast niedoszły przebój rynkowy sir Clifa Sinclair'a — **QL**, pomimo stosowania mikroprocesora MOTOROLA 68008.

Niestety, brak jest na liście niektórych mikrokomputerów, które z pewnością by się na niej znalazły, gdyby poddano je testowi. Mam tu na myśli takie urządzenia, jak: **ZENITH** (różne modele), **APPLE, LISA, CROMEMCO** (różne modele) czy chociażby popularny **COMMODORE 64** (omawiany w numerze 5, 1983 PCW, lecz nie testowany na szybkość). Z tego względu, jak również z uwagi na ograniczony zakres testów, opublikowanej listy nie można traktować jako w pełni aktualnej, kompletnej i niepodważalnej klasyfikacji.

W tabeli zestawiono 12 początkowych pozycji z listy miesięcznika PCW oraz 14 dalszych, wybranych dla bardziej znanych u nas mikrokomputerów. Dla tych czternastu, po nazwie, podano w nawiasie pozycję zajmowaną na liście. Rok określa w przybliżeniu wiek modelu; jest to rok, w którym testowano urządzenie. Zazwyczaj pojawiało się ono na rynku około pół roku wcześniej.

Na użytek Czytelników, którzy chcieliby przetestować nie figurujące na tej liście mikrokomputery, podajemy zestaw programów testowych stosowanych m.in. prz PCW. Wszystkie są napisane w języku BASIC i każdy z nich zawiera wykonywaną 1000 razy pętlę.

Test 1 zawiera „pustą" instrukcję FOR...NEXT i określa szybkość jej realizacji. **Test 2** zawiera również 1000 pustych pętli, lecz realizowanych poprzez instrukcję skoku. W **teście 3** dodano trochę arytmetyki.
Test 4 realizuje te same instrukcje, lecz zamiast zmiennych w linii 140 użyto stałych.
Test 5 pozwala zmierzyć szybkość realizacji instrukcji wywołania „pustego" podprogramu (GO SUB).
Test 6 miał prawdopodobnie testować szybkość instrukcji DIM, lecz w takim razie nie został wymyślony zbyt szczęśliwie (instrukcja w linii 130 powinna być umieszczona w taki sposób, aby w istotny sposób wpływała na czas realizacji programu) i w zasadzie test 6 stanowi poszerzenie testu 5 o 5000 wykonań pustej pętli FOR. **Test 7** otrzymano z poprzedniego, dodając instrukcję przypisania realizowaną na elemencie małej tablicy. Wreszcie **test 8** pozwala określić szybkość obliczania podstawowych funkcji.

```
100 REM TEST 1
110 PRINT "S"
120 FOR K=1 TO 1000
130 NEXT K
140 PRINT "E"
150 END
```

```
100 REM TEST 2
110 PRINT "S"
120 K=0
130 K=K+1
140 IF K<1000 THEN 130
150 PRINT "E"
160 END
```

```
100 REM TEST 3
110 PRINT "S"
120 K=0
130 K=K+1
140 A=K/K*K+K-K
150 IF K<1000 THEN 130
160 PRINT "E"
170 END
```

```
100 REM TEST 4
110 PRINT "S"
120 K=0
130 K=K+1
140 A=K/2*3+4-5
150 IF K<1000 THEN 130
160 PRINT "E"
170 END
```

Używając tych programów przetestowano zakupiony przez Instytut Ekonometrii Uniwersytetu Łódzkiego mikrokomputer IMP-86, produkowany przez firmę IMPOL II. Jest to model kompatybilny z IBM PC i wykorzystuje oryginalne oprogramowanie IBM. Dla wszystkich testów otrzymaliśmy czasy nieco lepsze niż podane w czasopiśmie PCW dla IBM PC. Interesujące byłoby porównanie wyników testów innych mikrokomputerów produkowanych w kraju, np. MERITUM czy COMPAN.

Lista najszybszych mikrokomputerów według Personal Computer World, January 1985

Typ mikrokomputera	Rok testowania	Czasy realizacji poszczególnych testów (sekundach)								Średnia czasów
		T1	T2	T3	T4	T5	T6	T7	T8	
SAGE II	1983	0,50	0,70	1,30	1,70	2,10	5,10	6,40	18,00	4,47
CRYSTAL	1984	0,70	1,70	4,60	3,90	4,30	7,70	13,60	18,70	6,91
IBM AT	1984	0,80	2,20	4,90	5,10	5,60	9,40	15,00	13,90	7,11
OLIVETTI M24	1984	0,80	2,50	5,20	5,20	5,70	10,00	15,30	16,60	7,66
MAD1	1984	0,70	2,60	5,50	5,70	6,20	10,90	16,80	17,30	8,21
XEROX 16/8	1984	1,20	3,60	7,40	7,70	8,70	16,00	4,70	24,70	9,25
FUTURE FX-20	1983	1,20	3,40	6,80	7,00	7,90	14,50	22,20	17,50	10,06
TULIP	1983	1,00	3,70	6,00	6,10	7,80	15,50	23,20	17,50	10,11
HP 110	1984	0,65	3,40	6,60	6,80	7,70	14,20	21,80	21,90	10,38
JONOS	1984	0,80	2,50	6,70	6,70	7,20	12,70	20,10	34,40	11,39
ADVANCE 86	1984	1,10	3,50	7,50	7,60	8,30	14,90	23,20	26,10	11,53
APPLE MACINTOSH	1984	0,24	0,65	6,00	6,40	7,10	8,60	15,90	52,30	12,15
BBC B (19)	1982	1,00	3,10	8,20	8,70	9,10	13,90	21,40	51,00	14,55
AMSTRAD (20)	1984	1,09	3,28	9,16	9,61	10,20	19.03	30,18	34,20	14,59
SIRIUS 1 (21)	1982	1,80	5,30	10,70	11,10	12,90	24,20	37,10	27,90	16,38
APRICOT (23)	1983	1,60	5,20	10,60	11,00	12,40	22,90	35,40	24,40	16,69
IBM PC (26)	1981	1,50	5,20	12,10	12,60	13,60	23,50	37,40	35,00	17,61
QL (31)	1984	2,10	6,40	10,70	10,30	13,20	26,10	61,80	25,80	19,55
TRS-80 II (36)	1980	1,00	5,00	13,00	13,00	14,00	23,00	35,00	60,00	20,50
IBM PC JUNIOR (39)	1984	1,90	6,40	11,20	15,00	16,50	29,30	46,60	47,40	21,79
APPLE III (42)	1982	1,70	7,20	13,50	14,50	16,00	27,00	42,50	75,00	24,68
VIC-20 (48)	1981	1,40	8,30	15,50	17,10	18,30	27,20	42,70	99,00	28,69
APPLE II (49) ś	1979	1,30	8,50	16,00	17,80	19,10	28,60	44,80	107,00	30,39
ZX-81 (61)	1981	4,50	6,90	16,40	15,80	18,60	49,70	68,50	229,00	51,18
ZX-SPECTRUM (63)	1982	4,80	8,70	21,10	20,40	24,00	55,30	80,70	253,00	58,50
ATARI (65)	1980	2,30	7,40	19,90	23,20	26,80	40,70	61,50	431,00	76,60

```
100 REM TEST 5
110 PRINT "S"
120 K=0
130 K=K+1
140 A=K/2*3+4-5
150 GOSUB 190
160 IF K<1000 THEN 130
170 PRINT "E"
180 END
190 RETURN
```

```
100 REM TEST 6
110 PRINT "S"
120 K=0
130 DIM(5)
140 K=K+1
150 A=K/2*3+4-5
160 GOSUB 220
170 FOR L=1 TO 5
180 NEXT L
190 IF K<1000 THEN 140
200 PRINT "E"
210 END
220 RETURN
```

```
100 REM TEST 7
110 PRINT "S"
120 K=0
130 DIM M(5)
140 K=K+1
150 A=K/2*3+4-5
160 GOSUB 230
170 FOR L=1 TO 5
180 M(L)=A
190 NEXT L
200 IF K<1000 THEN 140
210 PRINT "E"
220 END
230 RETURN
```

```
100 REM TEST 8
110 PRINT "S"
120 K=0
130 K=K+1
140 A=K↑2
150 B=LOG(K)
160 C=SIN(K)
170 IF K<1000 THEN 130
180 PRINT "E"
190 END
```

Przy testach tego rodzaju należy pamiętać, że dotyczą one tylko jednej z cech mikrokomputera, a i to traktowanej fragmentarycznie. Zacytowany zestaw testów np. nie pozwala określić szybkości operacji graficznych na ekranie. Znaczenie danej cechy jest różne dla różnych użytkowników, w zależności od charakteru zastosowań i metod pracy. Gdy w grę wchodzą cechy, których nie obejmują cytowane programy, trzeba niestety samemu wymyśleć test. Jest to zazwyczaj o tyle trudne, że wkraczamy w obszar rozbieżności między różnymi wersjami języka BASIC.

RYSZARD KURZYJAMSKI
Uniwersytet Łódzki

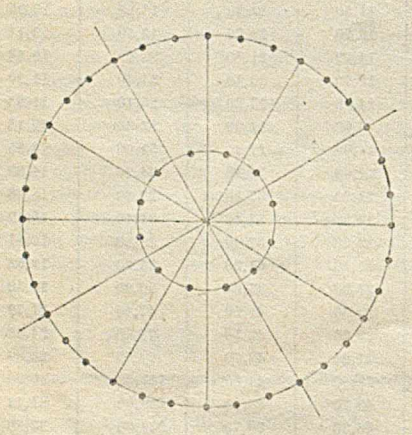
Hasło „metody numeryczne" wywołuje u wielu osób gęsią skórkę. Tymczasem zanim zdecydujemy się na przejście do programowania w języku ASSEMBLER warto czasem zajrzeć do podręcznika matematyki. Może się bowiem nagle okazać, że to, co przedłużało czas realizacji programu niemal w nieskończoność, można zrealizować inną metodą, znacznie szybszą i równie dokładną.

W zasadzie wszyscy wiedzą, że obliczanie funkcji trygonometrycznych trwa dłużej niż mnożenie. A jednak w niemalże wszystkich programach kreślących okręgi — współrzędne kolejnych punktów wyliczane są mozolnie z wykorzystaniem sinusa i cosinusa. Jak pokazuje poniższy tekst można to samo zrobić znacznie szybciej.

Przykładowe programy zostały zapisane w języku BASIC w wersji zaimplementowanej na komputerach AMSTRAD/SCHNEIDER CPC 464. Jednak nikomu nie powinno sprawić wiele kłopotu przeniesienie ich na dowolny inny komputer. AMSTRAD szczególnie nadaje się do zastosowań graficznych dzięki wysokiej rozdzielczości ekranu. Nie ma to jednak wpływu na algorytm przedstawionych procedur.

Sposoby kreślenia okręgów

W przedstawionych poniżej programach procedura kreśląca koło poprzedzona jest fragmentem umożliwiającym wprowadzenie z klawiatury danych, określających współrzędne środka okręgu, promień oraz liczby punktów, które będą wykorzystywane do jego wykreślania. Liczba punktów, wystarczająca do wykreślania okręgu (zakładając maksymalną dokładność przy danej rozdzielczości monitora lub sterownika CRT) związana jest z długością promienia okręgu i zwiększa się wraz z jego długością (rys. 1).



Rys. 1. Liczba punktów składających się w okrąg zależy od jego promienia

Okręgi o mniejszych promieniach można wykreślić wykorzystując mniejszą liczbę punktów. Zbyt duża liczba punktów niepotrzebnie wydłuża czas kreślenia, gdy przekroczony został zakres rozdzielczości [1]).

Przy kreśleniu okręgu zazwyczaj wykorzystywane są równania parametryczne. Dla okręgu o promieniu r i środku s(a,b) kolejne punkty można wyliczyć z następujących wzorów:

$$x_k = a + r \sin(k\Delta\varphi)$$
$$y_k = b + r \cos k\Delta\varphi)$$
$$0 \leqslant k\varphi\Delta \leqslant 360$$

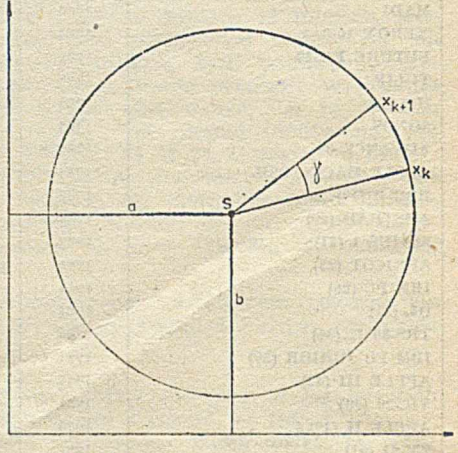
Przykładowy program wykorzystujący tę metodę przedstawia wydruk 1. W tym przykładzie zamiast liczby punktów należy podać kąt $\Delta\varphi$ określający co ile stopni będzie obliczany i rysowany punkt okręgu.

[1] Szczególnie wyraźnie widać to przy zmianie trybu rozdzielczości np. w komputerze AMSTRAD — przyp. AJP.

```
10 REM******************
20 REM
30 REM program 1
40 REM
50 REM******************
60 CLS
70 INPUT "wspolrzedne srodka"; a%, b%
80 INPUT "promien": r%
90 INPUT "kat"; d
100 REM*****************
110 REM
120 REM kreslenie okregu
130 REM
140 REM*****************
150 CLS
160 DEG
170 ORIGIN a%, b%
180 FOR k=0 TO 360 STEP d
190 x%=r%*COS(k) : y%=r%*SIN(k)
200 PLOT x%, y%
210 NEXT
```

Wydruk 1. Program 1

Wykorzystanie w programie równań parametrycznych nie jest rozwiązaniem szczególnie efektywnym. Jakkolwiek program kreślący jest bardzo prosty, wymaga jednak dla każdego punktu okręgu wykonania przez kom-



Rys. 2. Okrąg w układzie współrzędnych. Parametry a i b określają współrzędne środka okręgu s

puter wyliczeń wartości funkcji sinus i cosinus, co powoduje, że kreślenie okręgu odbywa się stosunkowo wolno.

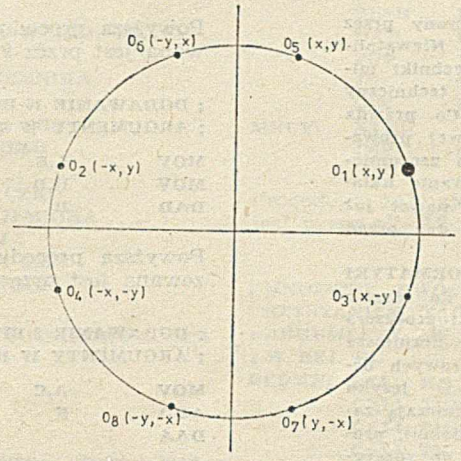
Opierając się na znanych wzorach trygonometrycznych można otrzymać równania, dla których wystarczy tylko raz obliczyć wartości funkcji sinus i cosinus dla kąta γ. Kąt γ otrzymujemy w wyniku podzielenia 360° (pełny okrąg) przez liczbę punktów, które chcemy wykorzystać do wykreślenia okręgu. Kolejne punkty okręgu wyznaczane są iteracyjnie według następujących wzorów (rys. 2).

$$x_{k+1} = a + (x_k - a) \quad \cos\gamma - (y_k - b) \quad \sin\gamma$$
$$y_{k+1} = b + (x_k - b) \quad \sin\gamma + (y_k - a) \quad \cos\gamma$$

Uwaga: sin γ i cos γ — wartości stałe $\gamma = \frac{360}{N}$, gdzie N określa liczbę punktów okręgu. Opisana metoda znacznie przyspiesza kreślenie okręgu. Przykładem wykorzystania tej metody jest PROGRAM 2 (wydruk 2).

```
10 REM******************
20 REM
30 REM program 2
40 REM
50 REM******************
60 CLS
70 INPUT "wspolrzedne srodka"; a%, b%
80 INPUT "promien"; r%
90 INPUT "liczba punktow"; n%
100 REM*****************
110 REM
120 REM kreslenie okregu
130 REM
140 REM*****************
150 CLS
160 DEG
170 ORIGIN a%, b%
180 p=360/n%
190 w1=COS(p):w2=SIN(p)
200 xp=r%:yp=0
210 FOR k%=1 TO n%
220 x=xp*w1—yp*w2
230 y=xp*w2+yp*w1
240 PLOT x, y
250 xp=x:yp=y
260 NEXT
```

Wydruk 2. Program 2



Rys. 3. Dodatkowe siedem punktów, które można uzyskać metodą prostych operacji mając wyliczone współrzędne jednego punktu

```
10 REM******************
20 REM
30 REM program 3
40 REM
50 REM******************
60 CLS
70 INPUT "wspolrzedne srodka"; a%, b%
80 INPUT "promien"; r%
90 INPUT "kat"; d
100 REM*****************
110 REM
120 REM kreslenie okregu
130 REM
140 REM*****************
150 CLS
160 DEG
170 ORIGIN a%, b%
180 n=360/8
190 FOR k=0 TO n STEP d
200 x%=r%*COS(k):y%=r%*SIN(k)
210 PLOT x%, y%
220 PLOT —x%, y%
230 PLOT x%,—y%
240 PLOT —x%,—y%
250 PLOT y%, x%
260 PLOT —y%, x%
270 PLOT y%,—x%
280 PLOT —y%,—x%
290 NEXT
```

Wydruk 3. Program 3

```
10 REM******************
20 REM
30 REM program 4
40 REM
50 REM******************
60 CLS
70 INPUT "wspolrzedne srodka"; a%, b%
80 INPUT "promien"; r%
90 INPUT "liczba punktow"; n%
100 REM*****************
110 REM
120 REM kreslenie okregu
130 REM
140 REM*****************
150 CLS
160 DEG
170 ORIGIN a%, b%
180 p=360/n%
190 m%=n%/8
200 w1=COS(p):w2=SIN(p)
210 xp=r%:yp=0
220 FOR k%=1 TO m%
230 x=xp*w1—yp*w2
240 y=xp*w2+yp*w1
250 xp=x:yp=y
260 x%=x:y%=y
270 PLOT x%, y%
280 PLOT —x%, y%
290 PLOT x%,—y%
300 PLOT —x%,—y%
310 PLOT y%, x%
320 PLOT —y%, x%
330 PLOT y%,—x%
340 PLOT —y%,—x%
350 NEXT
```

Wydruk 4. Program 4

Czas kreślenia okręgu pierwszą lub drugą metodą można jeszcze kilkakrotnie skrócić wykorzystując symetrię okręgu. Wyznaczenie jednego punktu pozwala w bardzo prosty sposób określić dodatkowo współrzędne dalszych siedmiu punktów okręgu, bez konieczności wykonywania dla nich obliczeń funkcji trygonometrycznych (PROGRAM 1) lub operacji mnożenia (PROGRAM 2, rys. 3). Uwzględniając w obliczeniach symetrię okręgu, PROPRAM 1 oraz PROGRAM 2 zmienią się odpowiednio w PROGRAM 3 i PROGRAM 4 (wydruk 3 i 4).

MAREK DORYWALSKI
Przemysłowy Instytut Informatyki
Warszawa

Po okresie prosperity firmy produkujące mikrokomputery przeżywają kryzys. Na rynku angielskim jedynie AMSTRAD wychodzi „na swoje", wypracowując zysk w wysokości 20 milionów funtów. Z ogłoszonych pod koniec roku raportów wynika, że firmy takie jak COMMODORE czy ACORN ponoszą codziennie straty rzędu... miliona funtów. Niewiele lepiej wygląda sytuacja firmy SINCLAIR. Wprawdzie komputerów QL i SPECTRUM+ sprzedaje się dwa razy więcej niż produktów firmy COMMODORE i ACORN razem wziętych, to jednak SINCLAIR zmuszony był zwolnić 20 (spośród 120) pracowników. Sprzedawcy zgodzili się na przejęcie nagromadzonych w magazynach zapasów SPECTRUM+ i QL tylko pod warunkiem opóźnienia promocji nowego modelu komputera SPECTRUM 128 na rynku angielskim aż do lutego 1986. Sir Clive Sinclair — szef firmy — nie mógł jednak oprzeć się pokusie pokazania światu nowego produktu. SPECTRUM 128 składane przez firmę Investronica właśnie pojawiło się w... Hiszpanii. Spodziewają się go też niektóre sklepy w RFN.

Hiszpańskie SPECTRUM zawiera ten sam układ ULA, co dotychczasowy model. Sugeruje to, że możliwości graficzne SPECTRUM 128 nie zostały poprawione. Jest to sprzeczne z plotkami o nowej wersji mikrokomputera. Plotki te wynikły z cech prototypów dostarczonych firmom produkującym oprogramowanie. Jest więc prawdopodobne, że na rynek angielski trafi nieco inna wersja SPECTRUM.

•

Rozczarowani nabywcy komputera AMSTRAD CPC 664 zyskali sprzymierzeńca w firmie DK'tronics, która do tej pory znana była z przystawek do ZX SPECTRUM. Oferuje ona za 50 funtów przystawkę dodającą 64 KB RAM do pamięci CPC 664. Podobnie jak w przypadku AMSTRADA CPC 6128 dodatkowa pamięć nie może być wykorzystywana przez interpreter języka BASIC. Firma DK zapowiedziała też kolejną przystawkę z pamięcią 256 KB za cenę 100 funtów i tzw. dysk na krzemie (pamięć podtrzymywana bateryjnie) również o pojemności 256 KB.

Odpowiednik mikroprocesora 8080 produkowany przez CEMI, znajduje coraz większe zastosowanie. Niewątpliwie należy się cieszyć z upowszechniania techniki mikroprocesorowej, choć szkoda, że postęp techniczny opiera się na tak archaicznym układzie. Co prawda pracownicy Instytutu Technologii Elektronowej prowadzą dyskretny sondaż środowiska, dając do zrozumienia, że niebawem wezmą się za opracowywanie układów bardziej nowoczesnych. Gdyby jednak nawet już dziś zapadły decyzje na tzw. szczeblu, to i tak przed nami jeszcze wiele lat oczekiwania.

Mamy nadzieję, że proces wydawniczy INFORMATYKI nie będzie aż tak długi, toteż oddajemy do druku pierwszą część zestawu procedur dla mikroprocesora 8080. Zdając sobie sprawę z tego, że coraz liczniejsza grupa użytkowników systemów mikroprocesorowych będzie musiała zająć się programowaniem w języku ASSEMBLER (niewielkie możliwości 8080 ograniczają zakres wykorzystywania języków wyższego poziomu), proponujemy procedury o stosunkowo prostym do rozgryzienia algorytmie. Procedury optymalizowane pod kątem czasu realizacji lub miejsca zajmowanego w pamięci zawsze zawierają pewne haczyki, które ograniczają możliwości uniwersalnego stosowania. Zacznijmy więc od prostszych przykładów.

Przykłady procedur w języku ASSEMBLER 8080 (1)

W programach użytkowych, realizujących wyznaczone przez projektanta funkcje, często powtarzają się typowe procedury lub ciągi instrukcji. Należą do nich m.in. podprogramy realizujące niektóre operacje arytmetyczne, funkcje elementarne, operacje na tablicach, macierzach itp. Tego typu procedury zbierane są w biblioteki, fragmenty których użytkownik może dołączyć do swojego programu. Operacje mniej skomplikowane, z uwagi na szeroki ich zestaw nie są łączone w biblioteki, co oczywiście nie oznacza, że konstruktor powinien zawsze wymyślać je od początku. Każdy programista ma oczywiście własną bibliotekę prostych procedur, w których wykorzystuje typową dla siebie sekwencję instrukcji i sposób przekazywania parametrów. Dzięki stosowaniu wypróbowanych wzorców projektant może koncentrować swoją uwagę na optymalnym zapisaniu algorytmu, bez drobiazgowej analizy zawartości poszczególnych rejestrów podczas wykonywania typowych operacji.

Poniżej podajemy przykładowy zestaw prostych programów, realizujących stosunkowo często występujące operacje.

PROGRAMY ARYTMETYCZNE

W prostych systemach realizowanych na mikroprocesorze 8080 argumentami operacji arytmetycznych są zazwyczaj liczby 8- lub 16-bitowe. Mogą to być wartości odczytane z przetworników analogowo-cyfrowych, parametry nastaw, wartości zadane przez operatora itp. Argumenty przedstawiane są z reguły w kodzie binarnym lub BCD. W poniższych programach przyjęto, że argumenty znajdują się w rejestrach C i E (8 bitów) lub parach rejestrów BC i DE (16 bitów), natomiast wynik działania w rejestrze A (8 bitów) lub parze rejestrów HL (16 bitów).

Dodawanie

Dodawanie jest operacją najprostszą, z uwagi na to, że procesor 8080 ma w swoim zestawie rozkazów operacje dodawania liczb 8- i 16-bitowych. Oto zapis przykładowych procedur:

```
; DODAWANIE 8-BITOWYCH LICZB BINARNYCH
MOV     A, C        ; POBIERZ PIERWSZY SKŁADNIK
ADD     E           ; DODAJ DRUGI SKŁADNIK
```

Powyższa procedura zajmuje w pamięci 2 bajty i realizowana jest przez 9 cykli zegara (4,5 μs).

```
; DODAWANIE 16-BITOWYCH LICZB BINARNYCH
; ARGUMENTY W REJ. BC i DE; WYNIK W REJ. HL
MOV     L,E         ; POBIERZ PIERWSZY SKŁADNIK
MOV     H,D
DAD     B           ; DODAJ DRUGI SKŁADNIK
```

Powyższa procedura zajmuje w pamięci 3 bajty i realizowana jest przez 20 cykli zegara (10 μs).

```
; DODAWANIE 8-BITOWYCH LICZB BCD
; ARGUMENTY W REJ. C I E; WYNIK W REJ. A
MOV     A,C         ; POBIERZ PIERWSZY SKŁADNIK
ADD     E           ; DODAJ DRUGI SKŁADNIK
DAA                 ; KOREKCJA DZIESIĘTNA
```

Powyższa procedura zajmuje w pamięci 3 bajty i realizowana jest przez 13 cykli zegara (6,5 μs)

```
; DODAWANIE 16-BITOWYCH LICZB BCD
; ARGUMENTY W REJ. BC i DE; WYNIK W REJ. HL
MOV   A,C   ; POBIERZ LSB PIERWSZEGO SKŁADNIKA
ADD   E     ; DODAJ LSB DRUGIEGO SKŁADNIKA
DAA         ; KOREKCJA BCD LSB
MOV   L,A   ; ZAPAMIĘTAJ LSB SUMY
MOV   A,B   ; POBIERZ MSB PIERWSZEGO SKŁADNIKA
ADC   D     ; DODAJ MSB DRUGIEGO SKŁADNIKA BINARNIE
DAA         ; KOREKCJA BCD MSB
MOV   H,A   ; ZAPAMIĘTAJ MSB SUMY
```

Powyższa procedura zajmuje w pamięci 8 bitów i realizowana jest przez 36 cykli zegara (18 μs).

Odejmowanie

Procesor 8080 nie realizuje rozkazu odejmowania liczb 16-bitowych i dlatego tę operację należy wykonać jako dwukrotne odejmowanie liczb 8-bitowych. W przypadku liczb BCD należy znaleźć dopełnienie odjemnika do 100 i dodać do odjemnej.

```
; ODEJMOWANIE 8-BITOWYCH LICZB BINARNYCH
; ARGUMENTY W REJ. C I E; WYNIK W REJ. A
MOV     A,C         ; POBIERZ ODJEMNĄ
SUB     E           ; ODJEMNA W REJ. E, RÓŻNICA W REJ. A
```

Procedura zajmuje w pamięci 2 bajty i realizowana jest przez 9 cykli zegara (4,5 μs).

```
; ODEJMOWANIE 16-BITOWYCH LICZB BINARNYCH
; ARGUMENTY W REJ. BC I DE; WYNIK W REJ. HL
MOV     A,C         ; LSB ODJEMNEJ
SUB     E           ; ODEJMIJ LSB ODJEMNIKA
MOV     L,A         ; LSB RÓŻNICY
MOV     A,B         ; MSB ODJEMNEJ
SBB     D           ; ODEJMIJ MSB ODJEMNIKA Z POŻYCZKĄ
MOV     H,A         ; MSB RÓŻNICY
```

Procedura zajmuje w pamięci 6 bajtów i realizowana jest przez 38 cykli zegara (19 μs).

```
; ODEJMOWANIE 8-BITOWYCH LICZB BCD
; ARGUMENTY W REJ. C i E; WYNIK W AKUMULATORZE
MVI     A, 9AH      ; DOPEŁNIENIE DO 99 BEZ POŻYCZKI
SUB     E           ; DOPEŁNIENIE ODJEMNIKA
ADD     C           ; DODAJ ODJEMNĄ
DAA                 ; KOREKCJA DZIESIĘTNA WYNIKU
```

Procedura zajmuje w pamięci 5 bajtów i realizowana jest przez 15 cykli zegara (7,5 μs).

```
; ODEJMOWANIE 16-BITOWYCH LICZB BC
; ARGUMENTY W REJ. BC I DE ; WYNIK W REJ. HL
MVI     A, 9AH  ; DOPEŁNIENIE DO 100
SUB     E       ; DOPEŁNIENIE LSB ODJEMNIKA
ADD     C       ; DODAJ ODJEMNĄ
DAA             ; KOREKCJA DZIESIĘTNA LSB
MOV     L,A     ; ZAPAMIĘTAJ LSB WYNIKU
MVI     A, 99H  ; DOPEŁNIENIE DO 99
ACI     0       ; UWZGLĘDNIENIE POŻYCZKI
SUB     D       ; DOPEŁNIENIE MSB ODJEMNIKA
ADD     B       ; DODAJ MSB ODJEMNEJ
DAA             ; KOREKCJA DZIESIĘTNA MSB
MOV     H,A     ; ZAPAMIĘTAJ MSB WYNIKU
```

Procedura zajmuje w pamięci 14 bajtów i realizowana jest przez 55 cykli zegara (27 μs).

Mnożenie

Wszystkie operacje mnożenia, wykonywane przez procesor 8080 muszą być realizowane w sposób programowy. Mnożenie liczb całkowitych 8- i 16-bitowych bez znaku może być wykonane na rejestrach procesora, co znacznie skraca ich czas wykonania. W pierwszym przedstawionym przykładzie realizowane jest mnożenie liczb 8-bitowych (lub liczby 8-bitowej i 16-bitowej) dające wynik 16-bitowy (bez sprawdzania, czy wynik mnożenia mieści się w szesnastu bitach).

Drugi przykład przedstawia mnożenie dwóch liczb 16-bitowych, dające wynik 32-bitowy.

Trzeci przykład przedstawia mnożenie dwóch dwucyfrowych liczb BCD. Celem przyspieszenia operacji przedstawiono realizację algorytmu wykorzystującego tabliczkę mnożenia. Niestety takie rozwiązanie znacznie zwiększa zajmowany przez program obszar pamięci.

```
; MNOŻENIE 8-BITOWYCH LICZB BINARNYCH (LUB LICZBY
; 16-BITOWEJ PRZEZ 8-BITOWĄ)
; MNOŻNA W REJ. C (PARZE REJ. BC); MNOŻNIK W REJ. E
MNOZ:   MVI B, 0        ; ZEROWANIE DLA LICZBY 8-BITOWEJ
DMNOZ:  LXI H,0         ; ILOCZYN POCZĄTKOWY 0
        MVI D,8         ; LICZNIK BITÓW
        MOV A,E         ; MNOŻNIK W AKUMULATORZE
MNOZ1:  DAD H           ; PRZESUŃ WYNIK POŚREDNI
        RLC             ; NAJSTARSZY BIT MNOŻNIKA
        JNC MNOZ2
        DAD B           ; BIT MNOŻNIKA = 1, DODAJ MNOŻNĄ
MNOZ2:  DCR D           ; UWZGLĘDNIJ 8 BITÓW
        JNZ MNOZ1
        RET
```

Procedura zajmuje w pamięci 12 bajtów i realizowana jest przez ok. 75 μs.

```
; MNOŻENIE LICZB 16-BITOWYCH, WYNIK 32-BITOWY
; MNOŻNA W REJ. BC; MNOŻNIK W REJ. DE;
; ILOCZYN W REJ. DE (MSB) I HL (LSB)
MNOZ16: MVI     A, 10H ; LICZNIK BITÓW
        LXI     H,0    ; WARTOŚĆ POCZĄTKOWA
        PUSH    H      ; WYNIK NA STOS
MNBIT:  XCHG
        DAD     H      ; NAJSTARSZY BIT DO CY
        XCHG
        JNC     NDOD ; NIE DODAWAJ, JEŚLI CY = 0
        DAD     B      ; DO LSB DODAJ MNOŻNĄ
        JNC     NDOD ; JEST PRZENIESIENIE ?
        XTHL
        INX     H      ; UWZGLĘDNIJ PRZENIESIENIE W MSB
        XTHL
NDOD:   DCR     A      ; WSZYSTKIE BITY ?
        JNZ     MMSB
        POP     D      ; POBIERZ MSB
        RET
;
MMSB:   DAD     H      ; PRZESUŃ LSB W LEWO
        XTHL           ; POBIERZ MSB
        PUSH    PSW ; ZACHOWAJ PRZENIESIENIE Z LSB
        DAD     H      ; PRZESUŃ MSB
        POP     PSW
        JNC     MNCY ; BRAK PRZENIESIENIA Z LSB ?
        INX     H      ; DODAJ PRZENIESIENIE DO MSB
MNCY:   XTHL           ; MSB NA STOS
        JMP     MNBIT
```

Procedura zajmuje w pamięci 37 bajtów i realizowana jest przez ok. 2,2 ms.

```
; MNOŻENIE DWÓCH LICZB 2-CYFROWYCH BCD, Z WYKO-
; RZYSTANIEM TABLICZKI MNOŻENIA. WYNIK 4-CYFROWY.
; ARGUMENTY W REJ. 8 (WZ) I REJ. C (XY) ; WYNIK
; W REJ. HL
BCDMN:  LXI  H,0    ; WYNIK POŚREDNI
        CALL TMP    ; HL + W*X
        CALL TM10   ; 10*HL + X*Z
        CALL TM     ; HL + Y*W
        CALL TM10   ; 10*HL + Y*Z
        RET
;
TM10:   DAD  H      ; PRZESUŃ WYNIK POŚREDNI
        DAD  H      ; O JEDNĄ CYFRĘ
        DAD  H
        DAD  H
TMP:    MOV  A,B    ; PRZYGOTUJ MNIEJ ZNACZĄCY BAJT
        RLC         ; DO WYCIĘCIA
        RLC
        RLC
        RLC
        MOV  B,A
```

```
; PODPROGRAM MNOŻENIA Z TABLICZKĄ
TM:     MOV  A,B
        ANI  OFH    ; PRAWA CYFRA PIERWSZEGO CZYNNIKA
        MOV  E, A
        MOV  A,C
        ANI  0F0H   ; LEWA CYFRA DRUGIEGO CZYNNIKA
        ORA  E      ; POŁĄCZ CYFRY
        PUSH H      ; SCHOWAJ WYNIK CZĘŚCIOWY
        LXI  D,TABMN
        MOV  L,A    ; PRZYGOTUJ ADRES
        MVI  H,0    ; DLA TABLICZKI MNOŻENIA
        DAD  D
        MOV  A,M    ; ILOCZYN Z TABLICZKI
        POP  H
        ADD  L      ; DODAJ ILOCZYN CZĘŚCIOWY
        DAA         ; KOREKCJA BCD
        MOV  L,A
        MVI  A,0
        ADC  H      ; UWZGLĘDNIJ PRZENIESIENIE
        DAA         ; KOREKCJA BCD
        MOV  H,A    ; BARDZIEJ ZNACZĄCE CYFRY WYNIKU
        MOV  A,B    ; USTAW CYFRY
        MOV  B,C    ; DLA KOLEJNEGO
        MOV  C,A    ; WYCIĘCIA
        RET
```

```
; TABLICZKA MNOŻENIA
TABMN:  DB      00,00,00,00,00,00,00,00,00.....
        DB      00,01,02,03,04,05,06,07,08.....
        ......
        DB      00,09,18H,27H,36H,45H,54H.....
```

Procedura zajmuje w pamięci 56 bajtów, tabliczka mnożenia 144 bajty — razem 200 bajtów. Czas realizacji ok. 0,75 ms.

Dzielenie

Jako przykład operacji dzielenia przedstawione zostanie dzielenie dwóch liczb 16-bitowych. Wynikiem dzielenia jest 16-bitowy iloraz oraz 16-bitowa reszta.

```
; DZIELENIE 16-BITOWYCH LICZB BINARNYCH
; DZIELNA W REJ. BC ; DZIELNIK W REJ. DE,
; ILORAZ W REJ. BC, RESZTA W REJ. HL
DZIEL:  MOV   A,D
        CMA           ; NEGACJA REJESTRU D
        MOV   D,A
        MOV   A,E
        CMA           ; NEGACJA REJESTRU E
        MOV   E,A
        INX   D       ; UZUPEŁNIENIE DZIELNIKA W U2
        LXI   H,0     ; WARTOŚĆ POCZĄTKOWA RESZTY
        MVI   A,17    ; LICZNIK PĘTLI
DZIEL1: PUSH  H       ; ZAPAMIĘTAJ WARTOŚĆ RESZTY
        DAD   D       ; ODEJMIJ DZIELNIK
        JNC   DZIEL2  ; MIEŚCI SIĘ JEŻELI CY = 1
        XTHL          ; RÓŻNICA NA STOS
DZIEL2: POP   H       ; POBIERZ AKTUALNĄ RESZTĘ
        PUSH  PSW     ; ZAPAMIĘTAJ LICZNIK PĘTLI
        MOV   A,C
        RAL           ; PRZESUŃ DZIELNĄ W LEWO O JEDEN
                      BIT
        MOV   C,A     ; NA BIT NAJMNIEJ ZNACZĄCY
                      ; WPROWADŹ
        MOV   A,B     ; WARTOŚĆ ILORAZU
        RAL
        MOV   B,A
        MOV   A,L     ; PRZESUŃ RESZTĘ O JEDEN BIT
                      ; W LEWO
        RAL           ; NA BIT NAJMNIEJ ZNACZĄCY
                      ; WPROWADŹ
        MOV   L,A     ; NAJBARDZIEJ ZNACZĄCY BIT DZIELNEJ
        MOV   A,H
        RAL
        MOV   H,A
        POP   PSW     ; ODTWÓRZ LICZNIK PĘTLI
        DCR   A
        JNZ   DZIEL1
        DRA   A       ; KASUJ BIT CY
        MOV   A, H
        RAR           ; RESZTA PRZESUNIĘTA
        MOV   H,A     ; O JEDEN BIT ZA DALEKO
        MOV   A,L
        RER
        MOV   L,A
        RET
```

Procedura zajmuje w pamięci 44 bajty i jest realizowana przez ok. 1,3 ms.

W drugiej części artykułu przedstawione zostaną przykładowe procedury realizujące funkcje logiczne.

ADAM PLUTA
Przemysłowy Instytut Elektroniki
Warszawa

Nie chcemy, aby publikacje w mikroKLANIE adresowane do użytkowników konkretnych komputerów zostały zdominowane przez określony typ. Przeglądając dotychczasowe publikacje trudno oprzeć się wrażeniu lansowania ZX SPECTRUM. Nie wynika to jednak z preferencji redakcji. Jest to obecnie najpopularniejszy typ mikrokomputera w naszym kraju.

Ostatnio pojawił się nowy kandydat na gwiazdę — CPC 464. Poświęciliśmy mu już nieco miejsca, lecz mimo że polecamy go wielu potencjalnym użytkownikom, przygotowujemy również publikację, która odsłoni słabsze punkty tej konstrukcji (być może ukaże się ona już w usamodzielnionym mikroKLANIE).

Poniżej publikujemy krótki tekst praktyczny — dedykowany użytkownikom tego komputera. Ich liczba wzrasta, o czym może świadczyć zalążek powstającego w Warszawie klubu użytkowników AMSTRADA/SCHNEIDERA.

Zapamiętywanie obrazu w CPC 464

W mikrokomputerze AMSTRAD (SCHNEIDER) CPC 464 można zarejestrować w pamięci masowej obraz z ekranu wykorzystując w tym celu instrukcję SAVE. Instrukcja SAVE z opcją B dla tego typu zbioru (ang. Binary) pozwala zapisać na dysku lub kasecie zawartość obszaru pamięci w postaci binarnej. Parametrami instrukcji są m.in. adres początkowy i wielkość obszaru (jeżeli zapisywany byłby program, możliwe jest podanie jeszcze adresu startu). Pamięć obrazu zajmuje przestrzeń adresową o rozmiarach 16 K bajtów (4000H), znajdującą się w obszarze o adresach 0C000H... ...OFFFFH.

Instrukcja SAVE w postaci:

SAVE "!"+„FILENAME",B, &C000, & 4000

zapamiętuje na dysku lub kasecie aktualny obraz z ekranu, jako zbiór binarny. Wyświetlenie wcześniej zapamiętanego obrazu realizuje instrukcja LOAD w postaci:

LOAD "!"+„FILENAME"

Pierwszym elementem ciągu znaków podawanych w zapisie instrukcji jest wykrzyknik (zapis "!"+„FILENAME" jest równoważny z zapisem „! FILENAME"). Nie jest on jednak pierwszym znakiem faktycznej nazwy zbioru, gdyż jako znak specjalny jest zawsze usuwany. W przypadku pracy z pamięcią dyskową nie ma on żadnego znaczenia i w podanych instrukcjach można go pominąć.

W przypadku pracy z kasetami wykorzystanie znaku „!" pozwala uniknąć wykorzystywania ekranu do zapisu lub odczytu wszelkiego typu komunikatów i pytań dotyczących np.: znalezionych, ładowanych, zapisywanych zbiorów, numerów bloków i czynności operatorskich. Informacje tego typu zwykle są niepożądane w zapamiętanym lub odczytanym obrazie. Z tego samego powodu instrukcje zapisu i odczytu należy umieścić w programie (w trybie konwersacyjnym na obraz nałoży się zapis instrukcji).

Zapamiętywany obraz zajmuje nieco więcej niż 16 K bajtów, dlatego też należy upewnić się, czy dysponujemy wystarczająco dużym obszarem pamięci masowej.

Wygodniejszym urządzeniem do przechowywania informacji jest oczywiście dysk elastyczny, gdyż umożliwia szybki zapis i odczyt zbiorów. W przypadku pracy z kasetą szybkość zapisu można zwiększyć, przez wykonanie instrukcji SPEED WRITE 1. Przy odczycie ustalenie odpowiedniej szybkości następuje w sposób automatyczny (na podstawie bitów synchronizacji). Zwiększenie szybkości zapisu-odczytu wymaga stosowania kaset wyższej jakości.

Posługując się instrukcjami SAVE i LOAD w przedstawionej formie (z wykrzyknikiem) należy pamiętać o włączeniu REC i PLAY — w przypadku zapisu na kasecie — oraz PLAY — w przypadku odczytu — gdyż komunikaty żądające tych czynności operatorskich nie pojawią się na ekranie.

MAREK DORYWALSKI
Warszawa

μK

prowadzi:
Andrzej J. Piotrowski
tel. dom.: 48-22-85